NASA-CR-192748

CIRSSE-R-63

NAGW-1333

(NASA-CR-192748) COLLISION AVOIDANCE OF MOBILE ROBOTS IN NON-STATIONARY ENVIRONMENTS (Rensselaer Polytechnic Inst.) 34 p

N93-71641

Unclas

Z9/63 0153774

Center for Intelligent Robotic Systems for Space Exploration

Rensselaer Polytechnic Institute
Troy, New York 12180-3590

COLLISION AVOIDANCE OF MOBILE ROBOTS IN NON-STATIONARY ENVIRONMENTS

by

K.J. Kyriakopoulos and G.N. Saridis

Rensselaer Polytechnic Institute
Electrical, Computer, and Systems Engineering
Troy, New York 12180-3590

September, 1990

CIRSSE REPORT #63

Collision Avoidance of Mobile Robots in Non-stationary Environments

K.J.Kyriakopoulos and G.N.Saridis

NASA Center for Intelligent Robotic Systems for Space Exploration
Robotics and Automation Laboratories

Rensselaer Polytechnic Institute
Troy, NY 12180-3590

abstract

A control strategy for real-time collision avoidance of a mobile robot in an environment containing moving obstacles is proposed. A dynamic model of the robot, the constraints and assumptions are presented. Objects, including the robot, are modelled as convex polyhedra. Collision avoidance is guaranteed if the minimum distance between the robot and the objects is nonzero. A nominal trajectory is assumed to be known from off-line planning. The main idea is to change the velocity along the nominal trajectory so that collisions are avoided. Furthermore, consistency with the nominal plan is desirable. The process is formulated as an optimization problem and a close to optimal solution is obtained. Simulation results verify the value of the proposed strategy.

1 Introduction

The trend to install several autonomous machines within the same environment raises questions about their interaction with their human "associates", and between themselves. Naturally, the issue of safety and efficient cooperation surfaces. Although a potential solution to this problem is priority assignement by a central control unit, this would contradict the obvious design specification regarding maximal "autonomy".

Therefore, a solution where a central control unit assigns only simple tasks specified by qualitative requirements, leaving the lower level decision making to be performed within the conceptual (or physical) limits of an autonomous machine, is desirable.

A typical case is this of an industrial floor (fig 1) and a mobile robot that was assigned the simple task to go, for example, from an initial configuration with orientation θ_A at the cartesian position A to a final orientation θ_B, and position B, within some time T, satisfying certain constraints and optimizing over certain criteria.

A **nominal motion plan** is determined based on the apriori knowledge of the environment. This knowledge is not guaranteed to be accurate because changes happen and **unexpected objects** enter the workspace. If these disrupt the motion of the robot, then the nominal motion plan must be modified. For example, in the setup of fig. 1 a collision is likely to happen when the moving obstacle is crossing the path of the mobile robot, at the neighborhood of point C.

The problem of treating moving objects has been stated as early as 1984 [FH84] and usually ad hoc solutions were given [Le89] [Tou86]. Reif and Sharir [RS85] gave an algorithmic solution to the problem but they were restricted to some categories of shapes of objects. Additionally, their approach is not suitable for an on-line implementation that is actually necessary in a dynamic environment. On the other hand, Kant and Zucker [KZ84], [KZ86], [KZ88], used the decomposition of the motion planning problem to the *find-path*, and *move-along-path* problems, they propose that the avoidance of moving obstacles can be done by adjusting the motion along the geometric path. The same approach was adopted in [WJ88], and recently in [GE90]. The basic idea of this approach is utilized in this work. Our scheme is more general and complete in the sense that the dynamic model of the robot is used, the objects are modelled as convex polyhedra and, in addition to collision avoidance, time consistency with the nominal plan is sought. Lately search based approaches for solving the above problem have also been presented in [FS90],[SLG90].

The unexpected objects must be detected, classified as disrupting or not, and if necessary, avoided. The first two issues have been presented in detail in the paper [KS90], while the **main issue** of this paper, is the avoidance of the special class of unpredictable moving objects that disrupt the motion of the mobile robot during only **a finite period of time**.

The basic idea of the proposed approach is the parametrization of the trajectory of the nominal motion plan. This is done by describing the shape and the orientation of the trajectory of the mobile robot by a function $r(s) = [p_x(s)\ p_y(s)\ \theta(s)]^T$ where s is a scalar variable in the interval $[0, s_f]$. The algorithm presented in this paper determines $s(t)$ so that collision avoidance is achieved. By introducing such a parametrization, the dimensionality of the problem becomes smaller, since by moving along $r(s)$ collision avoidance is guaranteed in the static environment.

In section 2 the problem, the assumptions and the constraints are stated. In section 3 the proposed strategy is presented. The numerical issues of the algorithms are discussed in section 4. Finally in section 5 simulation results are presented and discussed.

2 Problem Statement

We assume that a mobile robot is following a nominal plan (computed off-line) that is composed of a description $r(s)$ of the cartesian trajectory, and the motion function $s_n(t)$, $t \in [0, T]$. At every instant the t the position and orientation of the robot is given by $r(t) = r(s_n(t))$. If at a moment ($t = t_0$), a collision is predicted to happen at $t_c \in [t_0, T]$ (see [KS90]) a new motion function $s(t)$, $t \in [t_0, t_f]$, different from $s_n(t)$ should be found to guarantee collision avoidance and a final time (task execution time) t_f , as close as possible to the time T of the nominal plan.

An investigation of the dynamics of the mobile robot, the solid modelling of the obstacles and the robot and, finally, the assumptions for this scenario are presented in the sequel, thus enabling a more mathematical formulation of the above problem.

2.1 Mobile Robot Dynamic Modelling

The assumptions made for the dynamic modelling of the mobile robot are:

- The rotation velocity and steering angle of all the wheels satisfy the rolling compatibility conditions [AM89], thus avoiding **slipping** .
- The translational velocity $v(t) = \dot{s}(t)$ at any point s of a trajectory $r(s)$ is bounded by $v_{max}(s)$ so that the inertial forces do not saturate the available friction between the wheels and the floor, and therefore **skidding** is avoided. An approach to finding $v_{max}(s)$ is presented in [K.K90].
- The rotational kinematic energy of the rotating wheels is not considered. This is a realistic assumption since the wheels usually have considerably less mass than the whole robot.

Consider the dynamic energy of the moving robot:

$$K = \frac{1}{2} \cdot I \cdot \omega^2 + \frac{1}{2} \cdot m \cdot \dot{s}^2 \tag{1}$$

where I is the moment of inertia of the robot, and m is its mass. The first term corresponds to the rotational and the second to the translational motion.

From the description $r(s) = [x(s)\ y(s)\ \theta(s)]^T$ of the trajectory, the curvature

$$f(s) = \pm \frac{x' \cdot y'' - y' \cdot x''}{[x'^2 + y'^2]^{\frac{3}{2}}} = \theta'(s) \tag{2}$$

can be found. Obviously,

$$\omega(s) = \dot{s} \cdot f(s) \tag{3}$$

and therefore (1) becomes

$$K(s, \dot{s}) = \frac{1}{2} \cdot I \cdot f^2(s) \cdot \dot{s}^2 + \frac{1}{2} \cdot m \cdot \dot{s}^2 \tag{4}$$

The Lagrangian of the mechanical system of the robot is

$$L(s, \dot{s}) = K - P = K \tag{5}$$

and therefore the Euler-Lagrange equations, assuming that nonconservative forces such us friction are not present, are:

$$\frac{\partial}{dt} \frac{\partial L}{\partial \dot{s}} - \frac{\partial L}{\partial s} = u \tag{6}$$

where $-U_2 \leq u \leq U_1$ is the driving force of the robot. U_1 is the maximum thrust force that the actuators can give, while $-U_2$ is the maximum deccelation that can be applied based on the friction coefficient between the wheels of the robot and the floor.Using (4)

$$(m + I \cdot f^2(s)) \cdot \ddot{s} + I \cdot f(s) \cdot f'(s) \cdot \dot{s}^2 = u. \tag{7}$$

is obtained.

The second order dynamic model suggests the use of the phase plane, as an approporiate tool because phase plane can both represent the constraints and provide visualization of the state trajectory. To do that (7) is rewritten as

$$\frac{dt}{ds}(s) = \frac{1}{v(s)} \tag{8}$$

$$\frac{dv}{ds}(s) = -\frac{I \cdot f(s) \cdot f'(s)}{m + I \cdot f^2(s)} \cdot v(s) + \frac{1}{(m + I \cdot f^2(s)) \cdot v(s)} \cdot u(s) \tag{9}$$

2.2 Dynamic - Kinematic Constraints

i) In order that (8) be well defined $v(s) \geq \epsilon > 0$ where ϵ is a very small constant. Furthermore, the mobile robot is assumed to move with ideal rolling. Rolling without skidding is achieved as shown in [K.K90] by bounding the velocity $v(s)$. Thus

$$\epsilon \leq v(s) \leq v_{max}(s) \tag{10}$$

ii) Collision avoidance is guaranteed if the distance [GJ85] between the mobile robot and every object of the surrounding environment is greater than a safety positive constant d^o, i.e when

$$d(s,t) = \min_{i,j}\{\|z_i - z_j\| : \quad z_i \in C_r(s), z_j \in C_o(t)\} \geq d^o \qquad \forall s, t \tag{11}$$

where

$$C_r(s) = \{x / x \in \Re^3 \ni A_r \cdot R_r^{-1}(s) \cdot x \leq b_r - A_r \cdot R_r^{-1}(s) \cdot T_r(s), \quad A_r \in \Re^{m \times 3}, b_r \in \Re^m\}, \tag{12}$$

$$C_o(t) = \{y / y \in \Re^3 \ni A_o \cdot R_o^{-1}(t) \cdot y \leq b_o - A_o \cdot R_o^{-1}(t) \cdot T_o(t), \quad A_o \in \Re^{l \times 3}, b_o \in \Re^l\}, \tag{13}$$

are **convex polyhedra** representing the convex hulls of the mobile robot and the moving obstacle respectively. (A_r, b_r) and (A_o, b_o) are the parameters that define the convex polyhedron description of the robot and the object respectively, with respect to their fixed coordinate frame. R_r,R_o,T_r,T_o represent the rotation and translation of the frames of the robot and the object with respect to the world frame. A computationally efficient approach to estimate $d(s,t)$ and predict the collision time t_c under uncertain input from sensing devices is presented elsewhere [KS90]. The collision time is defined by

$$t_c = \inf_{t \in [t_0, t_0 + T_h]}\{t / d(t) = 0\} \tag{14}$$

where t_0 is the present time, and T_h is the time horizon of interest.

2.3 Assumptions

In order to minimize the on-line information processing, certain assumptions have to be made:

i) A **nominal plan** is available from off-line motion planning and is described by:

- $r(s) = [p_x(s)\ p_y(s)\ \theta(s)]^T$, describing the shape of the cartesian trajectory and the orientation angle along it.

- $s_n(t)$ describing the motion with respect to time along the trajectory $r(s)$, where $0 \le t \le T$, $s_n(0) = s_0$ and $s_n(T) = s_f$.(See fig.2)

- $v_{max}(s)$ giving at point s the maximum velocity in order to satisfy the non-skidding and stability constraints. Additionally feasibility in terms of control input u, when following $v_{max}(s)$ has to be guaranteed. (see fig.2)

- $t_{min}(s) = \int_0^s \frac{1}{v_{max}(\sigma)} \cdot d\sigma$ giving the minimum time from $(s_0 = s_n(t_0), v_{max}(s_0))$ to $(s, v_{max}(s))$.

ii) **Temporary Obstruction Assumption**: The mobile robot moving along path $r(s)$ can be obstructed during only a bounded amount of time i.e the moving object is assumed not to permanently stay on, or move parallely to $r(s)$.

2.4 Performance Criterion

In this paper, the only imposed performance criterion is time consistency with respect to the initial plan. This is expressed by minimizing

$$J = (t(s_f) - T)^2 \tag{15}$$

where $t(s_f)$ is the arrival time at s_f, and T is the time to reach s_f, according to the nominal plan.

2.5 Mathematical Problem Statement

Based on the previous discussion, the mathematical statement of the problem of collision avoidance of moving obstacles is now straightforward. It is reminded that s is the trajectory parametrization variable.

system

$$x'(s) = A(x(s)) + B(x(s)) \cdot u(s) \tag{16}$$

where $(\cdot)' = \frac{d(\cdot)}{ds}$, $x(s) = [t(s)\ v(s)]^T$, with $v(s) = \frac{ds}{dt}(s)$,

$$A(x) = \begin{bmatrix} \frac{1}{v(s)} \\ \frac{-l \cdot f(s) \cdot f'(s)}{m + l \cdot f^2(s)} \cdot v(s) \end{bmatrix} \in \Re^{2\times 1} \tag{17}$$

and

$$B(x) = \begin{bmatrix} 0 \\ \frac{1}{m+I\cdot f^2(s)} \cdot \frac{1}{v(s)} \end{bmatrix} \in \Re^{2\times 1} \tag{18}$$

initial-final conditions

$$x(s_0) = [0\ v_0]^T \qquad x(s_f) = [\text{free}\ v_f]^T \tag{19}$$

input constraints

$$-U_2 \leq u(s) \leq U_1 \tag{20}$$

state constraints

$$\epsilon \leq v(s) \leq v_{max}(s) \tag{21}$$

collision avoidance constraints

$$d_0 - d(s,t) \leq 0 \tag{22}$$

performance criterion

$$J = (t(s_f) - T)^2 \tag{23}$$

Minimizing (23) subject to eqs(16)-(22) using an Optimal Control Strategy is a non-trivial task. Is is also a time consuming process when real time computing is required.

In this paper, a Minimum Interference Strategy (MIS) is proposed for providing fast but in general suboptimal solutions to the above problem. The suboptimality results from the fact that eq. (22) is approximated and not treated directly. The development of a computationally efficient process for an optimal solution of eq. (23), or other criteria, is the subject of on-going research. However, since any optimal control solution is going to be obtained numerically, using some iterative process, the solution provided by MIS may serve as a first guess to speed up the numerical computations.

3 Minimum Interference Strategy

In the sequel some definitions, lemmas and theorems that establish the proposed Minimum Interference Strategy are presented. The relevant proofs are presented in Appendix A.

Lemma 3.1: The following sets:

$$D = \{(s,t)/d(s,t) \leq 0\} \tag{24}$$

$$D_t = \{t/\ \exists s\ \ni\ (s,t) \in D\} \tag{25}$$

$$D_s = \{s/\ \exists t\ \ni\ (s,t) \in D\} \tag{26}$$

are non-empty and all of them are compact.

The following theorem mathematically establishes the proposed strategy.

Theorem 3.2: The *collision avoidance conditions* are

$$s(t) \geq s_2 = \max D_s \qquad \forall t \geq t_1 = \min D_t \tag{27}$$

$$s(t) \leq s_1 = \min D_s \qquad \forall t \leq t_2 = \max D_t \tag{28}$$

Conditions (27) and (28) provide two collision avoidance options. The first condition provides avoidance by acceleration and called **Accelerating Minimum Interference Strategy** (AMIS), while the second condition provides avoidance by decceleration and called **Deccelerating Minimum Interference Strategy** (DMIS). Their implementation must be such that the overall time $t(s_f)$ is as close as possible to T, the total time according to the nominal plan.

Furthermore (27) and (28) are quite general but their use, in the general case, becomes problematic because of the difficulties in calculating s_1, s_2, t_1, t_2. Difficulties arise from the nonlinear, in general, form of $r(s)$ and the nondifferentiable nature of $d(s,t)$. If

$$\rho = \frac{\delta_r + \delta_o}{2} \tag{29}$$

where δ_r, δ_o are the diameters of the convex polyhedra C_r, C_o , then "tubes" with radii ρ can be set around curves $r_p(s) = [p_x(s)\ p_y(s)]$ and $o(t) = [o_x(t)\ o_y(t)]$ describing the actual and predicted cartesian translational trajectories of the robot and the moving obstacle respectively. From the intersections of these "tubes" with $r(s)$ and $o(t)$, safe estimates of s_1, s_2, t_1, t_2 can be derived. This is demonstrated on fig. 3.

Definition 3.3 : Consider the functional $t : \Re \times \Re \times C[s_0, s_f] \rightarrow \Re$ defined by

$$t(s_1, s_2, v) = \int_{s_1}^{s_2} \frac{1}{v(s)} \cdot ds \tag{30}$$

representing the time needed to go from point s_1 to point s_2 moving following a velocity function function $v(s)$.

Lemma 3.4.a (b) : The minimum (maximum) velocity v_{min}^{0i} (v_{max}^{0i}) that the mobile robot system eqs.(16- 21) can achieve at point s_i, starting from $[s_0\ v_0]^T$ where $s_i > s_0$ and moving exactly t_j seconds (assuming that t_j is enough time for this motion), is obtained following a velocity trajectory $v_c(s)$ composed of:

- an arc with input $u_c(s) = U_1 \qquad (= -U_2)$,
- an arc with $v_c(s) = v_{max}(s) \qquad (= v_{min}(s) = \epsilon)$, and
- an arc with input $u_c(s) = -U_2 \qquad (= U_1)$

Any one of the above arcs may not exist.

Lemma 3.5.a(b) : The maximum (minimum) velocity v_{max}^{if} (v_{min}^{if}) with which the mobile robot system eqs.(16- 21) can start from point s_i leading to $[s_f\ v_f]^T$ ($s_i < s_f$) and moving exactly $T - t_j$ seconds (assuming that $T - t_j$ is enough time for this motion), is obtained following a velocity trajectory $v_c(s)$ composed of:

- an arc with input $u_c(s) = -U_2 \qquad (= U_1)$,
- an arc with $v_c(s) = v_{min}(s) \qquad (= v_{max}(s))$, and

- an arc with input $u_c(s) = U_1 \qquad (= -U_2)$

Any one of the above arcs may not exist.

Lemma 3.6 : Given a time t_j, the corresponding velocities defined in lemmas 3.4.(a,b), 3.5.(a,b) are monotonous functions of t_j and
i) $v^{0i}_{min}(t_j)$ is decreasing function of t_j
ii) $v^{0i}_{max}(t_j)$ is decreasing function of t_j
iii) $v^{if}_{min}(t_j)$ is increasing function of t_j
iv) $v^{if}_{max}(t_j)$ is increasing function of t_j .

MIS is implemeted in two phases. In phase I, the space of feasible velocities at s_i $i = 1,2$ is determined, and a strategy (AMIS or DMIS) is chosen. In phase II, the actual trajectory is specified.

3.1 MIS Phase I : Feasibility Study and Strategy Selection

Phase I of MIS is actually a feasibility test. Parameters s_1, s_2, t_1, t_2 that specify the "dangerous segment-time" area are inputs to this phase. During this phase, decision is made upon which of the two strategies, AMIS or DMIS, is going to be selected. Depending on the selected strategy the following parameters are determined and provided to Phase II of MIS, which is the trajectory planning stage.

- DMIS : $(s_1, t_2, t_f, v^1_{max} = \min(v^{01}_{max}, v^{1f}_{max}), v^1_{min} = \max(v^{01}_{min}, v^{1f}_{min}))$
- AMIS : $(s_2, t_1, t_f, v^2_{max} = \min(v^{02}_{max}, v^{2f}_{max}), v^2_{min} = \max(v^{02}_{min}, v^{2f}_{min}))$

The above parameters are iteratively found. The issue of convergence of these iterative schemes is of major significance in this work, and is covered in section 4.

ALGORITHM: MIS Phase I

STEP 1: Feasiblity of Deccelerating MIS: Based on Lemma 3.4.b, the maximum time t^{01}_{max} from (s_0, v_0) to s_1 is found.

if $t^{01}_{max} < t_2$
then DMIS-NOT-FEASIBLE:=1; goto STEP 2
else goto STEP 3.

STEP 2: Feasiblity of Accelerating MIS:

1) The minimum time t^{02}_{min} from (s_0, v_0) to s_2 is determined.

2) Based on Lemma 3.4.a v^{02}_{min} is found.

3) Integrate backwards (9) with final conditions (s_f, v_f) and $u(s) = -U_2$ until $s = s_2$. Record $v(s_2)$.

if If $t^{02}_{min} > t_1$ or $v(s_2) < v^{02}_{min}$
then AMIS-NOT-FEASIBLE:=1

STEP 3: Strategy Selection:

if AMIS-NOT-FEASIBLE=1 **and** DMIS-NOT-FEASIBLE=1
then *"Collision Unavoidable"*
else if DMIS-NOT-FEASIBLE:=0
then goto STEP 4
else goto STEP 5

STEP 4: The Deccelerating MIS

- Based on Lemmas 3.4.a,b, an interval $I^{01} = [v^{01}_{min}, v^{01}_{max}]$ of feasible velocities at point s_1 is obtained. Every $v^{01} \in I^{01}$ at s_1, can be achieved in exactly time t_2.
- Based on Lemma 3.5.a(b) an interval $I^{1f} = \left[v^{1f}_{min}, v^{1f}_{max}\right]$ of feasible velocities at point s_1 is obtained. Therefore v_f at s_f can be achieved starting from every $v^{1f} \in I^{1f}$ at s_1, within exactly time $T - t_2$.
- Interval $I^1 = [v^1_{min}, v^1_{max}] = I^{01} \cap I^{1f}$ is found. If:
  - $I^1 \neq \emptyset$: Then one feasible arc from (s_0, v_0) to $(s_1, v(s_1) \in I^1)$, and one from $(s_1, v(s_1) \in I^1)$ to (s_f, v_f) can be easily constructed. Obviously the total time $t_f = T_{DMIS} = T$; **goto PHASE II**
  - $v^{01}_{min} > v^{1f}_{max}$ then:
    - **Increase** time t_2 until $v^{01}_{min} = v^{1f}_{max}$.(Based on Lemma 3.6)
    - set $I^1 = v^{01}_{min} = v^{1f}_{max}$.
    - Obviously the total time $t_f = T_{DMIS} = T$; **goto PHASE II**
  - $v^{01}_{max} < v^{1f}_{min}$ then:
    - set $I^1 = v^{01}_{max}$
    - construct the **maximum time arc** from (s_0, v_0) to (s_1, v^{01}_{max}).
    - Record the total time $T_{DMIS} > T$.
    - Goto 2 to check feasibility of the Accelerating MIS;
      - if AMIS-NOT-FEASIBLE = 1 , **goto PHASE II**
      - else goto STEP 5.

STEP 5: The Deccelerating MIS

- Based on Lemmas 3.4.a,b, an interval $I^{02} = [v^{02}_{min}, v^{02}_{max}]$ of feasible velocities at point s_2 is obtained. Every $v^{02} \in I^{02}$ at s_2, can be achieved in exactly time t_1.
- Based on Lemma 3.5.a(b) an interval $I^{2f} = \left[v^{2f}_{min}, v^{2f}_{max}\right]$ of feasible velocities at point s_2 is obtained. Therefore the final velocity v_f at s_f can be achieved starting from every $v^{2f} \in I^{2f}$ at s_2, within exactly time $T - t_1$.
- Interval $I^2 = [v^2_{min}, v^2_{max}] = I^{02} \cap I^{2f}$ is found. If:
  - $I^2 \neq \emptyset$: Then one feasible arc from (s_0, v_0) to $(s_2, v(s_2) \in I^2)$, and one from $(s_2, v(s_2) \in I^2)$ to (s_f, v_f) can be easily constructed. Obviously the total time $t_f = T_{AMIS} = T$; **goto PHASE II**
  - $v^{02}_{max} < v^{2f}_{min}$ then:
    * **Decrease** time t_1 until $v^{02}_{max} = v^{2f}_{min}$. (Based on Lemma 3.6)
    * set $I^2 = v^{02}_{max} = v^{2f}_{min}$.
    * Obviously the total time $t_f = T_{AMIS} = T$; **goto PHASE II**
  - $v^{02}_{min} > v^{2f}_{max}$ then:
    * set $I^2 = v^{02}_{min}$
    * construct the **maximum time arc** from (s_2, v^{02}_{min}) to (s_f, v_f).
    * Record the total time $T_{AMIS} < T$.

STEP 6: Suboptimal Strategy Selection

if $(T_{DMIS} - T)^2 \leq (T_{AMIS} - T)^2$

then choose DMIS; $t_f = T_{DMIS}$;**goto PHASE II**

else choose AMIS; $t_f = T_{AMIS}$;**goto PHASE II**

3.2 MIS Phase II : Trajectory Planning

Assuming that one of DMIS, AMIS has been selected in Phase I, the determination of a velocity function $v(s)$ along $r(s)$ that satisfies the input constraints (20) and the specifications $(s_i, t_j, t_f, v^i_{max}, v^i_{min})$ provided by the applied strategy is the remaining issue.

Any one of the two possible strategies determines a point $s_i \in [s_0, s_f]$ (s_1 or s_2), an interval $I^i = [v^i_{min}, v^i_{max}] \subset [\varepsilon, v_{max}(s_i)]$ of feasible velocities at s_i, the final time t_f ($= T_{AMIS}$ or T_{DMIS}), and a time instant $t_j \in [t_0, t_f]$ ($= t_1$ or t_2) at which the robot must be at s_i. Thus the problem can be divided in two subproblems with identical structure:
a) Starting from (s_0, v_0) find $v(s) \in C[s_0, s_i]$, corresponding to a feasible control input $u(s)$,

leading to $(s_i, v(s_i))$, where $v(s_i) \in I^i$, and satisfying $\int_{s_0}^{s_i} \frac{1}{v(s)} ds = t_j$.
b) Starting from $(s_i, v(s_i))$ find $v(s) \in C[s_i, s_f]$, corresponding to a feasible control input $u(s)$, leading to (s_f, v_f) and satisfying $\int_{s_i}^{s_f} \frac{1}{v(s)} ds = t_f - t_j$.

At this point, two issues have to be clarified. First, solutions to both subproblems (a) & (b) are guaranteed to exist from the feasibility study of Phase I. Second, Phase I only provides the segment I^i where $v(s_i)$ has to belong.

Accelerating MIS

The proposed structure of $v(s) \quad s \in [s_0, s_f]$ is (fig.4) :

- arc I with velocity $v(s) = v_n(s) \quad s \in [s_0, s_c]$
- arc II with input $u(s) = U_1 \quad s \in [s_c, s_v]$
- arc III with velocity $v(s) = v_{max}(s) \quad s \in [s_v, s_b]$
- arc IV with input $u(s) = -U_2 \quad s \in [s_v, s_2]$
- arc V with input $u(s) = -U_2 \quad s \in [s_2, s_d]$
- arc VI with velocity $v(s) = v_{min}(s) = \epsilon \quad s \in [s_d, s_u]$
- arc VII with input $u(s) = U_1 \quad s \in [s_u, s_r]$
- arc VIII with velocity $v(s) = v_n(s) \quad s \in [s_r, s_f]$

This gereral structure of $v(s)$ is going to be iteratively determined. In its final form some of the arcs may not appear.

Velocity $v(s_2)$ is selected as big as possible so that s_c is moved as close as possible to s_1, so that the "perturbation" is directed towards the future, when more sensing information is available. Variables $s_c, s_v, s_b, s_d, s_u, s_r$ are not independent but they are related because of

$$t(s_0, s_2, v) = \int_{s_0}^{s_2} \frac{1}{v(s)} \cdot ds = t_1 \tag{31}$$

$$t(s_2, s_f, v) = \int_{s_2}^{s_f} \frac{1}{v(s)} \cdot ds = t_f - t_1 \tag{32}$$

where $t(.,.,.)$ defined in (30). Therefore for a fixed $v(s_2)$ only two (e.g s_v, s_r) of those are independent. Those parameters are numerically obtained by the following algorithm.

ALGORITHM: Phase II AMIS

STEP 1: Determine iteration parameter:

Set $v(s_2) = v^2_{max} = \min(v^{02}_{max}, v^{2f}_{max})$

if $t(s_0, s_2, v) \geq t_1$

then goto STEP 2 (Find $v(s)$ $s \in [s_0, s_2]$)

else goto STEP 3 (Find $v(s)$ $s \in [s_0, s_2]$)

STEP 2: Iterate on s_v: Define $E(s_v) = \frac{1}{2} \cdot (t(s_v) - t_1)^2$

$s_v := s_2$

while ($E(s_v) > \varepsilon_0$, ε_0 : small)

$s_v := s_v - \gamma \cdot \frac{\frac{dE}{ds_v}}{\frac{d^2E}{ds_v^2}}$;

STEP 3: Iterate on $v_2 = v(s_2)$: Define $E(v_2) = \frac{1}{2} \cdot (t(v_2) - t_2)^2$

$v_2 = v(s_2)$

while ($E(v_2) > \varepsilon_0$, ε_0 : small)

$v_2 := v_2 - \gamma \cdot \frac{\frac{dE}{dv_1}}{\frac{d^2E}{dv_1^2}}$;

STEP 4: Find $v(s)$ $s \in [s_2, s_f]$: Define $E(s_r) = (t(s_r) - (t_f - t_1))^2$

$s_r = \frac{s_2 + s_f}{2}$

while ($E(s_r) > \varepsilon_0$, ε_0 : small)

$s_r := s_r - \gamma \cdot \frac{\frac{dE}{ds_r}}{\frac{d^2E}{ds_r^2}}$;

The numerical implementation issues of the above iterative schemes are discussed in the next section.

Deccelerating MIS

The proposed structure of $v(s)$ $s \in [s_0, s_f]$ is (fig.5) :

- arc I with velocity $v(s) = v_n(s)$ $s \in [s_0, s_c]$
- arc II with input $u(s) = -U_2$ $s \in [s_c, s_v]$
- arc III with velocity $v(s) = v_{min}(s) = v$ $s \in [s_v, s_b]$
- arc IV with input $u(s) = U_1$ $s \in [s_b, s_1]$
- arc V with input $u(s) = U_1$ $s \in [s_1, s_d]$
- arc VI with velocity $v(s) = v_{max}(s)$ $s \in [s_d, s_u]$
- arc VII with input $u(s) = -U_2$ $s \in [s_u, s_r]$
- arc VIII with velocity $v(s) = v_n(s)$ $s \in [s_r, s_f]$

The gereral structure of $v(s)$ is going to be iteratively determined. In its final form some of the arcs may not appear.

We propose that $v(s_1)$ is as small as possible so that s_c is moved as close as possible to s_1, so that the "perturbation" is directed towards the future, where more sensing information is available.

As in AMIS, variables $s_c, s_v, s_b, s_d, s_u, s_r$ are not independent but they are related from

$$t(s_0, s_1, v) = \int_{s_0}^{s_1} \frac{1}{v(s)} \cdot ds = t_2 \tag{33}$$

$$t(s_1, s_f, v) = \int_{s_1}^{s_f} \frac{1}{v(s)} \cdot ds = t_f - t_2 \tag{34}$$

Parameters s_v, s_r are numerically obtained by the following algorithm.

Phase II: DMIS

STEP 1: <u>Determine iteration parameter:</u>

Set $v(s_1) = v^1_{min} = \max(v^{01}_{min}, v^{1f}_{min})$

if $t(s_0, s_1, v) \leq t_2$

then goto STEP 2 (Find $v(s)$ $s \in [s_0, s_1]$)

else goto STEP 3 (Find $v(s)$ $s \in [s_0, s_1]$)

STEP 2: <u>Iterate on s_v</u>: Define $E(s_v) = (t(s_v) - t_2)^2$

$s_v := s_1$

while ($E(s_v) > \varepsilon_0$ ε_0 : small)

$s_v := s_v - \gamma \cdot \frac{\frac{dE}{ds_v}}{\frac{d^2E}{ds_v^2}}$;

STEP 3: <u>Iterate on $v_1 = v(s_1)$</u>: Define $E(v_i) = (t(v_i) - t_2)^2$

$v_1 = v(s_1)$

while ($E(v_i) > \varepsilon_0$ ε_0 : small)

$v_i := v_i - \gamma \cdot \frac{\frac{dE}{dv_i}}{\frac{d^2E}{dv_i^2}}$;

STEP 4: <u>Find $v(s)$ $s \in [s_0, s_1]$</u>: Define $E(s_r) = (t(s_r) - t_2)^2$

$s_r = \frac{s_1 + s_f}{2}$

while ($E(s_r) > \varepsilon_0$ ε_0 : small)

$s_r := s_r - \gamma \cdot \frac{\frac{dE}{ds_r}}{\frac{d^2E}{ds_r^2}}$;

The numerical implementation issues of the above iterative schemes are discussed in the next section.

4 Numerical Implementation Issues

Three computational problems are encountered in the Minimum Interference Strategy. Fortunately, they are well posed and solved with fast numerical techniques

i) Space of feasible velocities at s_i

In phase I of MIS, the following four problems have to be solved:

1)Based on Lemma 3.4.a find $v^{0i}_{max} = \max_{v(s)}\{v(s_i)/v(s) \in C[s_0, s_i] \ni t(s_0, s_i, v) = t_j\}$.

2)Based on Lemma 3.4.b find $v^{0i}_{min} = \min_{v(s)}\{v(s_i)/v(s) \in C[s_0, s_i] \ni t(s_0, s_i, v) = t_j\}$.

3)Based on Lemma 3.5.a find $v^{if}_{max} = \max_{v(s)}\{v(s_i)/v(s) \in C[s_i, s_f] \ni t(s_i, s_f, v) = T - t_j\}$.

4)Based on Lemma 3.5.b find $v^{if}_{min} = \min_{v(s)}\{v(s_i)/v(s) \in C[s_i, s_f] \ni t(s_i, s_f, v) = T - t_j\}$,

where $t(.,.,.)$ was defined in (30).

The use of lemmas 3.4, 3.5 enables the statement of the above problems as simple unidimensional optimization problems. For each of those problems a function

$$E_k = \frac{1}{2} \cdot (t_k(v_k) - T_k)^2 \tag{35}$$

is defined.$v_k \in \{v^{0i}_{max}, v^{0i}_{min}, v^{if}_{max}, v^{if}_{min}\}$, $T_k \in \{t_j, t_j, T - t_j, T - t_j\}$ where $k = 1, 2, 3, 4$ is the problem index. Every $t_k(v_k)$ is the time elapsed following a velocity arc with structure defined by the lemma corresponding to the problem. The sought solutions comes from minimizing (35). This is done using a *Newton Iteration*

$$v_k^{n+1} = v_k^n - \gamma \cdot \frac{\frac{dE_k}{dv_k}}{\frac{d^2E_k}{dv_k^2}} \tag{36}$$

where

$$\frac{dE_k}{dv_k} = (t_k(v_k^n) - T_k) \cdot \frac{dt_k(v_k^n)}{dv_k} \tag{37}$$

$$\frac{d^2E_k}{dv_k^2} = (\frac{dt_k(v_k^n)}{dv_k})^2 + (t_k(v_k^n) - T_k) \cdot \frac{d^2t_k(v_k^n)}{dv_k^2} \tag{38}$$

This numerical scheme converges geometrically to the optimal solution.

An analytic form for $\frac{dt_k(v_k^n)}{dv_k}$, $\frac{d^2t_k(v_k^n)}{dv_k^2}$ can be obtained by using the Leibniz's differentiation law for integrals. The analytic expressions of the above derivatives are given in appendix B.

ii) Evaluation of the parameters of the velocity arcs

In phase II of MIS. the following problems have to be solved:

1) In Step 2 of both AMIS and DMIS, a newton iteration is used to find s_v. To do so. $\frac{dE}{ds_v}, \frac{d^2E}{ds_v^2}$ where $E(s_v) = \frac{1}{2} \cdot (t(s_v) - t_j)^2$, must be known. Thus,

$$\frac{dE}{ds_v} = (t(s_v) - t_j) \cdot \frac{dt(s_v)}{ds_v} \tag{39}$$

$$\frac{d^2E}{ds_v} = (\frac{dt(s_v)}{ds_v})^2 + (t(s_v) - t_j) \cdot \frac{d^2t(s_v)}{ds_v^2} \tag{40}$$

Analytic expressions for $\frac{dt(s_v)}{ds_v}, \frac{d^2t(s_v)}{ds_v^2}$ are given in APPENDIX C-1.From the derivatives presented there, the fact that some additional assumptions have to be set so that the iterative scheme converges becomes obvious. The assumptions are:

- $v_{max}(s) - v_n(s) > \epsilon_v > 0$, $v_n(s) - v_{min}(s) > \epsilon_v > 0$ where ϵ_v very small constant.
- $v_{max}(s) \in C[s_0, s_f]$ (continuously differentiable).
- $u(s) = -I \cdot f(s) \cdot \frac{\partial f(s)}{\partial s} \cdot v^2_{max}(s) - (m + I \cdot f^2(s))v_{max}(s)\frac{\partial v_{max}(s)}{\partial s} < U_1 \quad s \in [s0, sf]$

2) In Step 3 of both AMIS and DMIS, a newton iteration is used to find v_i. To do that, $\frac{dE}{dv_i}, \frac{d^2E}{dv_i^2}$ where $E(v_i) = \frac{1}{2} \cdot (t(v_i) - t_j)^2$, must be known. Thus,

$$\frac{dE}{dv_i} = (t(v_i) - t_j) \cdot \frac{dt(v_i)}{dv_i} \tag{41}$$

$$\frac{d^2E}{dv_i} = (\frac{dt(v_i)}{dv_i})^2 + (t(v_i) - t_j) \cdot \frac{d^2t(v_i)}{dv_i^2} \tag{42}$$

Analytic expressions for $\frac{dt(v_i)}{dv_i}, \frac{d^2t(v_i)}{dv_i^2}$ are given in APPENDIX C-2.
3) In Step 3 of both AMIS and DMIS, a newton iteration is used to find s_r. This has the same structure as in case (ii-1) (search for s_v), and thus it is omitted.

iii) Feasible space update

An on-line scheme has to include an efficient update algorithm, to update the plan as new sensory information is recorded. Changes of the predicted motion of the moving obstacle [KS90] result in changes of s_1, s_2, t_1, t_2 defined in Thm. 3.2. According to Lemmas 3.4, 3.5 the new set s_1', s_2', t_1', t_2' creates changes of every $v_k \in \{v^{0i}_{min}, v^{0i}_{max}, v^{if}_{min}, v^{if}_{max}\}$ $i = 1, 2$.

A computationally efficient way to update $v_k(s_i, t_j)$ is done by expanding it in Taylor series, and then iterate in the way described in case (i). The Taylor series is written as

$$v_k(s_i', t_j') \approx v_k(s_i, t_j) + \frac{\partial v_k(s_i, t_j)}{\partial s_i} \cdot (s_i' - s_i) + \frac{\partial v_k(s_i, t_j)}{\partial t_j} \cdot (t_i' - t_i). \tag{43}$$

Analytic expressions for $\frac{\partial v_k(s_i,t_j)}{\partial s}, \frac{\partial v_k(s_i,t_j)}{\partial t}$ are given in APPENDIX D.

All iterations described above converge geometrically to the actual solution if γ is appropriately selected. Relevant analysis can be found in [BT89].

5 Simulation Results - Discussion

The environment of fig.1 was considered for simulation. Instead of treating a simple case of collision avoidance where the obtained trajectories have the form of fig.4,5, we preferred to test the efficiency of the proposed method by statistical analysis. In order to do this, 500 events where considered. Each event, is the appearance of a moving object nearby the robot. The velocities of the objects were uniformly distributed between $10 - 30m/s^2$. The mobile robot had mass of $50kg$, maximum thrust $110N$, maximum brakes force $-10N$, while the maximum velocity of the mobile robot was set to $8m/s$ for safety reasons. The random character of the events made the robot encounter extremely adverse situtations. For example, there were events in which,very speedy objects were suddenly "appearing" very close to the robot. This is, of course, not realistic.

The motion task time of the robot was initially $T = 9.02$ sec. From the 500 events, 390 could potentially create collisions. **Only DMIS** was simulated and gave collision free plans in 375 of them (96%).More interestingly, for those 375 cases that a collision free plan was succesfully determined, the distribution of the final time t_f (fig. 6) showed that it was close to T.

On going research has as a subject the development of real time optimal control schemes. Furthermore other possible criteria will be considered. Finally the proposed approach of moving along the same path, could be considered as too restrictive. A strategy that could probably deviate from this path should incorporate the nonholonomic constraints of the rolling motion.

APPENDIX A

I)*Proof of Lemma 3.1:* The proof is straightforward if the following facts are considered:
1) From (11)-(13) the fact that distance function $d(s,t)$ is a function of the rotation and translation matrices $(R_r(s), T_r(s), R_o(t), T_o(t))$, (i.e $d(s,t) = d(R_r(s), T_r(s), R_o(t), T_o(t)))$ becomes evident. In [GJ85] the distance functions were found to be continuous with respect to its arguments (R_r, T_r, R_o, T_o). But $R_r(s), T_r(s)$ and $R_o(t), T_o(t)$ are continuous functions of their arguments s, t. Therefore $d(s,t)$ is continuous with respect to (s.t).
2) A pair $(s_0 = s_n(t_0), t_0)$ exists such that $d(s_0, t_0) - d^0 > 0$ (meaning that prediction takes place before collision!).
3) The point (s_c, t_c) is finite and therefore only a bounded domain of $d(s,t)$ is considered here.Additionally, the definition of D shows that $D \subset \Re^2$, $D_t, D_s \subset \Re$ are all closed sets, and therefore compact.
4) A pair (s_c, t_c) exists such that $d(s_c, t_c) = 0$ and therefore $d(s_c, t_c) - d^0 < 0$ (t_c is determined by the Collision prediction strategy and given by (14), while $s_c = s_n(t_c)$). Continuity guarantees that D, D_t, D_s are nonempty. ∎

II)*Proof of Lemma 3.2:* The compactness of D, D_t, D_s guarantees the existence of the above minima and maxima. Obviously, the following relations are true.

$$s(t) \in D_s \Rightarrow t \notin D_t \tag{44}$$

and

$$t \in D_t \Rightarrow s(t) \notin D_s. \tag{45}$$

If the fact that $\dot{s}(t) > 0$ is considered, (27) and (28) are deduced.∎

III)*Proof of Lemma 3.4.a:* Consider an arc $v_c(s)$ (see fig 7) such that

$$t(s_0, s_i, v_c) = t_j \tag{46}$$

having the structure described above. Consider point s_b, where the arc corresponding to $u_c(s) = -U_2$ starts. For any other arc $v(s) \in C[s_0, s_i]$ corresponding to an input $u(s)$ satisfying (20), and such that

$$t(s_0, s_i, v) = t_j \tag{47}$$

we have that

$$v(s) \leq v_c(s) \quad \forall s \in [s_0, s_b] \tag{48}$$

because $v_c(s)$ is the maximum velocity arc starting from (s_0, v_0). Therefore

$$t(s_0, s_b, v_c) \leq t(s_0, s_b, v) \tag{49}$$

From (46, 47, 49)

$$t(s_b, s_i, v) \leq t(s_b, s_i, v_c) \tag{50}$$

is deduced.Assume that

$$v(s_i) < v_c(s_i) \tag{51}$$

From (49)

$$v(s_b) \leq v_c(s_b) \tag{52}$$

Equations (49- 52) suggest that there exist segments $I_k \subset [s_b, s_i]$ such that $v(s) > v_c(s) \quad \forall s \in I_k$. Consider

$$s_l = \sup_k\{s \in I_k\} \tag{53}$$

then obviously

$$\frac{dv}{ds}(s_l) < \frac{dv_c}{ds}(s_l) \quad \Rightarrow \tag{54}$$

$$u(s_l) < u_c(s_l) = -U_2 \tag{55}$$

The contradiction of (55) to (20) shows that assumption (51) is not valid. and therefore

$$v_{min}^{0i} = v_c(s_i) \leq v(s_i) \tag{56}$$

for all $v(s) \in C[s_0, s_i]$ corresponding to feasible controls $u(s)$.∎

The proof of lemmas 3.4.b, 3.5.a,b is similar to the above and omitted. The structure of the trajectories corresponding to both (3.4.a,b) is demonstrated in fig. 8, while for those in 3.5(a,b) in fig.9.

IV) *Proof of Lemma 3.6:* The structure of the arcs, defined by lemmas 3.4.(a,b), 3.5.(a,b), and leading to $v_{min}^{0i}, v_{max}^{0i}, v_{min}^{if}, v_{max}^{if}$ for a specific time t_j, shows that these functions are

1-1. Only (i) is proven here, since the others can be similarly proved. If $v^{0i}_{min}(t_j) = v(s_i)$, where $v(s)$ is an arc constructed for t_j according to lemma 3.4.a, is increased to a new value $v'(s_i)$, then the new arc $v(s)$ (according to lemma 3.4.a) leading to it, is going to be everywhere $v'(s) \geq v(s)$. Thus from (30) $t_j' \leq t_j$. Similarly, if $v^{0i}_{min}(t_j) = v(s_i)$, is decreased to a new value $v''(s_i)$, then the new arc $v(s)$ (according to lemma 3.4.a) leading to it, is going to be everywhere $v''(s) \leq v(s)$. Thus from (30) $t_j' \geq t_j$. QED ∎

APPENDIX B

The derivations of the following formulas is a tedious task. They are based on Leibniz's Diferentiation law for integrals. To help the better understanding of the derivation process, the relevant integrals and the corresponding figures are provided for every case. A typical procedure to derive those is provided for a relevant case in appendix C-2.

$\underline{k=1}$ (fig.8, line:2) $t_j = \int_{s_0}^{s_b} \frac{1}{v(s)} ds + \int_{s_b}^{s_i} \frac{1}{v(s)} ds$

$$\frac{\partial t_j}{\partial v^{0i}_{max}} = \frac{(m + I \cdot f^2(s_i))}{U_1} \cdot (1 - \frac{v^{0i}_{min}}{v(s_b)}) < 0 \tag{57}$$

$$\frac{\partial^2 t_j}{\partial (v^{0i}_{max})^2} = \frac{(m + I \cdot f^2(s_i))}{U_1} \cdot \frac{(m + I \cdot f^2(s_i))(v^{0i}_{max})^2 - (m + I \cdot f^2(s_b))v^2(s_d)}{v^3(s_b)} > 0 \tag{58}$$

$\underline{k=2}$ (fig.8, line:1) $t_j = \int_{s_0}^{s_b} \frac{1}{v(s)} ds + \int_{s_b}^{s_i} \frac{1}{v(s)} ds$

$$\frac{\partial t_j}{\partial v^{0i}_{min}} = \frac{(m + I \cdot f^2(s_i))}{U_1} \cdot (\frac{v^{0i}_{min}}{v(s_b)} - 1) < 0 \tag{59}$$

$$\frac{\partial^2 t_j}{\partial (v^{0i}_{min})^2} = \frac{(m + I \cdot f^2(s_i))}{U_2} \cdot \frac{(m + I \cdot f^2(s_b))v^2(s_b) - (m + I \cdot f^2(s_i))(v^{0i}_{min})^2}{v^3(s_b)} > 0 \tag{60}$$

$\underline{k=3}$ (fig.9, line:2) $T - t_j = \int_{s_i}^{s_d} \frac{1}{v(s)} ds + \int_{s_d}^{s_f} \frac{1}{v(s)} ds$

$$\frac{\partial (T - t_j)}{\partial v^{if}_{max}} = \frac{(m + I \cdot f^2(s_i))}{U_2} \cdot (1 - \frac{v^{if}_{min}}{v(s_d)}) < 0 \tag{61}$$

$$\frac{\partial^2 (T - t_j)}{\partial (v^{if}_{max})^2} = \frac{(m + I \cdot f^2(s_i))}{U_2} \cdot \frac{(m + I \cdot f^2(s_i))(v^{if}_{max})^2 - (m + I \cdot f^2(s_d))v^2(s_d)}{v^3(s_d)} > 0 \tag{62}$$

$\underline{k=4}$ (fig.9, line:1) $T - t_j = \int_{s_i}^{s_d} \frac{1}{v(s)} ds + \int_{s_d}^{s_f} \frac{1}{v(s)} ds$

$$\frac{\partial (T - t_j)}{\partial v^{if}_{min}} = \frac{(m + I \cdot f^2(s_i))}{U_1} \cdot (\frac{v^{if}_{min}}{v(s_d)} - 1) < 0 \tag{63}$$

$$\frac{\partial^2 (T - t_j)}{\partial (v^{if}_{min})^2} = \frac{(m + I \cdot f^2(s_i))}{U_2} \cdot \frac{(m + I \cdot f^2(s_d))v^2(s_d) - (m + I \cdot f^2(s_i))(v^{if}_{min})^2}{v^3(s_u)} > 0 \tag{64}$$

APPENDIX C

1)By applying Leibniz's Differentiation law for integrals, analytic expressions for $\frac{dt(s_v)}{ds_v}$, $\frac{d^2t(s_v)}{ds_v^2}$ are obtained.To help the better understanding of the derivation process, the relevant integrals and the corresponding figures are provided for every case.A typical procedure to derive those is provided in case 2 of this appendix.

As it can be determined from the following derivatives, the additional assumptions for convergence of section (4-ii) are necessary, so that local minima do not exist.

- For AMIS (fig.4) $t(s_v) = \int_{s_0}^{s_c} \frac{1}{v_n(s)} ds + \int_{s_c}^{s_v} \frac{1}{v(s)} ds + \int_{s_v}^{s_2} \frac{1}{v(s)} ds$
  - $v(s_v) < v_{max}(s_v)$ $(s_v \equiv s_b)$
    - $\frac{dt}{ds_v} = \frac{U_1+U_2}{U_1} \cdot (\frac{1}{v(s_c)} - \frac{1}{v(s_v)})$
    - $\frac{d^2t}{ds_v^2} = \frac{U_1+U_2}{U_1} \cdot (\frac{1}{v^3(s_c)} \frac{U_1+U_2}{m+I\cdot f^2(s_c)} - \frac{1}{v^3(s_v)} \frac{U_2}{m+I\cdot f^2(s_v)})$
  - $v(s_v) = v_{max}(s_v)$
    - $\frac{dt}{ds_v} = \frac{U_1-u(s_v)}{U_1} \cdot (\frac{1}{v(s_c)} - \frac{1}{v(s_v)})$
    - $\frac{d^2t}{ds_v^2} = -\frac{\frac{\partial u(s_v)}{\partial s}}{U_1} \cdot (\frac{1}{v(s_c)} - \frac{1}{v(s_v)}) + \frac{U_1-u}{U_1} \cdot (\frac{1}{v^3(s_c)} \frac{U_1-u}{m+I\cdot f^2(s_c)} - \frac{1}{v^2(s_v)} \frac{\partial v_{max}(s_v)}{\partial s})$

    where $u(s) = -I \cdot f(s) \cdot \frac{\partial f(s)}{\partial s} \cdot v^2(s) - (m + I \cdot f^2(s))v(s)\frac{\partial v_{max}(s)}{\partial s}$
- For DMIS (fig.5) $t(s_v) = \int_{s_0}^{s_c} \frac{1}{v_n(s)} ds + \int_{s_c}^{s_v} \frac{1}{v(s)} ds + \int_{s_v}^{s_2} \frac{1}{v(s)} ds$
  - $v(s_v) > v_{min}(s_v) = \varepsilon$ $(s_v \equiv s_b)$
    - $\frac{dt}{ds_v} = \frac{U_1+U_2}{U_2} \cdot (\frac{1}{v(s_c)} - \frac{1}{v(s_v)})$
    - $\frac{d^2t}{ds_v^2} = -\frac{U_1+U_2}{U_2} \cdot (\frac{1}{v^3(s_c)} \frac{U_1+U_2}{m+I\cdot f^2(s_c)} - \frac{1}{v^3(s_v)} \frac{U_1}{m+I\cdot f^2(s_v)})$
  - $v(s_v) = v_{max}(s_v)$
    - $\frac{dt}{ds_v} = \frac{U_1+I\cdot f(s_v)\cdot \frac{\partial f(s_v)}{\partial s}}{U_1} \cdot (\frac{1}{v(s_c)} - \frac{1}{v(s_v)})$
    - $\frac{d^2t}{ds_v^2} = \frac{I\cdot[(\frac{\partial f(s_v)}{\partial s})^2 + f(s_v)\frac{\partial^2 f(s_v)}{\partial s^2}}{U_2} \cdot (\frac{1}{v(s_c)} - \frac{1}{v(s_v)}) - \frac{U_2+I\cdot f(s_v)\cdot\frac{\partial f(s_v)}{\partial s}}{U_2} \cdot (\frac{1}{v^3(s_c)} \frac{U_1-u}{m+I\cdot f^2(s_c)})$

2) By applying Leibniz's Differentiation law for integrals, analytic expressions for $\frac{dt(v_1)}{dv_1}$, $\frac{d^2t(v_1)}{dv_1^2}$ are obtained. In order to demonstrate the derivation process the case of AMIS is analytically presented.

Consider fig.10. The arc has a first segment where $v(s) = v_n(s)$ $s \in [s_0, s_c]$, and a second segment resulting from an input $u(s) = U_1$ $s \in [s_c, s_i]$. The overall time is

$$t = \int_{s_0}^{s_c} \frac{1}{v_n(s)} ds + \int_{s_c}^{s_1} \frac{1}{v(s)} ds$$

Assuming that $v_1 = v(s_1)$ is an indepentent variable, then by Leibniz's Differentiation law for integrals,

$$\frac{dt}{dv_1} = -\int_{s_c}^{s_1} \frac{1}{v^2(s)} \cdot \frac{dv(s)}{dv_1} ds$$

However,

$$v^2(s) = \frac{m + I \cdot f^2(s_1)}{m + I \cdot f^2(s)} v^2(s_1) + \frac{2 \cdot U_1}{m + I \cdot f^2(s)}(s - s_1) \Rightarrow$$

$$\frac{dv(s)}{dv_1} = \frac{m + I \cdot f^2(s_1)}{m + I \cdot f^2(s)} \cdot \frac{v(s_1)}{v(s)}$$

Giving,

$$\frac{dt}{dv_1} = -(m + I \cdot f^2(s_1)) \cdot v(s_1) \cdot \int_{s_c}^{s_1} \frac{1}{v^3(s)} ds = \frac{m + I \cdot f(s_2)^2}{U_1} \cdot (1 - \frac{v(s_2)}{v(s_c)})$$

assuming small variation of $f(s)$ $s \in [s_c, s_1]$. The calculation of the second derivative is straightforward.

Following the same approach for DMIS, the following results were obtained:

- For AMIS
  - $\frac{dt}{dv_2} = \frac{m+I \cdot f(s_2)^2}{U_1} \cdot (1 - \frac{v(s_2)}{v(s_c)})$
  - $\frac{d^2t}{dv_2^2} = \frac{m+I \cdot f(s_2)^2}{U_1} \cdot \frac{(m+I \cdot f^2(s_2))v^2(s_2) - (m+I \cdot f^2(s_c))v^2(s_c)}{(m+I \cdot f^2(s_c))^2 v^3(s_c)}$
- For DMIS
  - $\frac{dt}{dv_1} = \frac{m+I \cdot f(s_1)^2}{U_2} \cdot (\frac{v(s_1)}{v(s_c)} - 1)$
  - $\frac{d^2t}{dv_1^2} = -\frac{m+I \cdot f(s_1)^2}{U_2} \cdot \frac{(m+I \cdot f^2(s_1))v^2(s_1) - (m+I \cdot f^2(s_c))v^2(s_c)}{(m+I \cdot f^2(s_c))^2 v^3(s_c)}$

APPENDIX D

By applying Leibniz's Differentiation law for integrals, analytic expressions for $\frac{\partial v_k(s_i,t_j)}{\partial s_i}$, $\frac{\partial v_k(s_i,t_j)}{\partial t_j}$ $v_k \in \{v_{min}^{0i}, v_{max}^{0i}, v_{min}^{if}, v_{max}^{if}\}$ $i = 1, 2$ are obtained:

$$\frac{\partial v_{max}^{0i}}{\partial s_i} = \frac{1}{(m + I \cdot f^2(s_i))} \cdot (\frac{U_1}{v_{max}^{0i} - v(s_b)} - I \cdot f(s_i) \cdot f'(s_i) \cdot v_{max}^{0i}) \tag{65}$$

$$\frac{\partial v_{max}^{if}}{\partial s_i} = \frac{1}{(m + I \cdot f^2(s_i))} \frac{1}{v_{max}^{if}} \cdot (\frac{U_2 \cdot v(s_d)}{v_{max}^{if} - v(s_d)} - U_2 - I \cdot f(s_i) \cdot f'(s_i) \cdot (v_{max}^{if})^2) \tag{66}$$

$$\frac{\partial v_{max}^{0i}}{\partial t_j} = \frac{1}{(m + I \cdot f^2(s_i))} \cdot (\frac{U_1 \cdot v(s_b)}{v_{max}^{0i} - v(s_b)}) \tag{67}$$

$$\frac{\partial v_{max}^{if}}{\partial t_j} = \frac{1}{(m + I \cdot f^2(s_i))} \cdot (\frac{U_2 \cdot v(s_d)}{v_{max}^{if} - v(s_d)}) \tag{68}$$

$$\frac{\partial v_{min}^{0i}}{\partial s_i} = \frac{1}{(m + I \cdot f^2(s_i))} \cdot (\frac{U_2}{v(s_b) - v_{min}^{0i}} - I \cdot f(s_i) \cdot f'(s_i) \cdot v_{min}^{0i}) \tag{69}$$

$$\frac{\partial v_{min}^{if}}{\partial s_i} = -\frac{1}{(m + I \cdot f^2(s_i))} \cdot (\frac{U_1}{v(s_d) - v_{min}^{if}} + I \cdot f(s_i) \cdot f'(s_i) \cdot (v_{min}^{if})^2) \tag{70}$$

$$\frac{\partial v_{min}^{0i}}{\partial t_j} = -\frac{1}{(m + I \cdot f^2(s_i))} \cdot \left(\frac{U_2 \cdot v(s_b)}{v(s_b) - v_{min}^{0i}}\right) \tag{71}$$

$$\frac{\partial v_{min}^{if}}{\partial t_j} = \frac{1}{(m + I \cdot f^2(s_i))} \cdot \left(\frac{U_1 \cdot v(s_d)}{v(s_d) - v_{min}^{if}}\right) \tag{72}$$

Acknowledgment

This work has been supported by the NASA Center for Intelligent Robotic Systems for Space Exploration (CIRSSE) under the NASA Grant NAGW-1333. The first author has been partially supported by a fellowship of Alexander Onasis Foundation.

References

[AM89] J. Alexander and J. Maddocks. On the kinematics of wheeled mobile robots. *International Journal of Robotics Research*, pages 15–27, October 1989.

[BT89] D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation.* Prentice Hall, New Jersey, 1989.

[FH84] E. Freund and H. Hoyer. Collision avoidance for industrial robots with arbitrary motion. *Journal of Robotic Systems*, 1984.

[FS90] K. Fujimura and H. Samet. Motion planning in a dynamic domain. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pages 324–330, 1990.

[GE90] N. Griswold and J. Eem. Control of mobile robots in the presence of moving objects. *IEEE Trans. Robotics and Automation*, pages 263–268, April 1990.

[GJ85] E. Gilbert and D. Johnson. Distance functions and their application to robot path planning in the presence of obstacles. *IEEE Journal of Robotics and Automation*, pages 21–30, March 1985.

[K.K90] K.Kyriakopoulos. A supervisory control strategy for navigation of mobile robots in dynamic environments - ph.d thesis proposal. Technical report, CIRSSE-RPI, February 1990.

[KS90] K. Kyriakopoulos and G Saridis. Minimum distance estimation and collision prediction under uncertainty for on-line robotic motion planning. In *Proceedings of the 1990 IFAC World Congress*, August 1990.

[KZ84] K. Kant and S. Zucker. Trajectory planning problems, i: Determining velocity along a fixed path. In *Proceeding of the IEEE 8th Int. Conf. Pattern Recogn.*, pages 196–198, 1984.

[KZ86] K. Kant and S. Zucker. Toward efficient trajectory planning: The path–velocity decomposition. *International Journal of Robotics Research*, pages 72–89, 1986.

[KZ88] K. Kant and S. Zucker. Planning collision-free trajectories in time-varying environments: A two-level hierarchy. In *Proceeding of the 1988 IEEE International Conference on Robotics and Automation*, pages 1644–1649, April 1988.

[Le89] Y. Liu and et.al. A practical algorithm for planning collision free coordinated motion of multiple mobile robots. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, pages 1427–1432, May 1989.

[RS85] J. Reif and M. Sharir. Motion planning in the presense of moving obstacles. Technical Report TR-06-85, Harvard University, Center for Research in Computing Technology, 1985.

[SLG90] C. Shih, T. Lee, and W. Gruver. Motion planning with time-varying polyhedral obstacles based on graph search and mathematical programming. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pages 331–337, 1990.

[Tou86] P. Tournassoud. A strategy for obstacle avoidance and its applications to multi-robot systems. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, pages 1224–1229, 1986.

[WJ88] C. Wu and C. Jou. Design of a controlled spatial curve trajectory for robot manipulators. In *Proceedings of the 27th Conference on Decision and Control*, pages 161–166, December 1988.

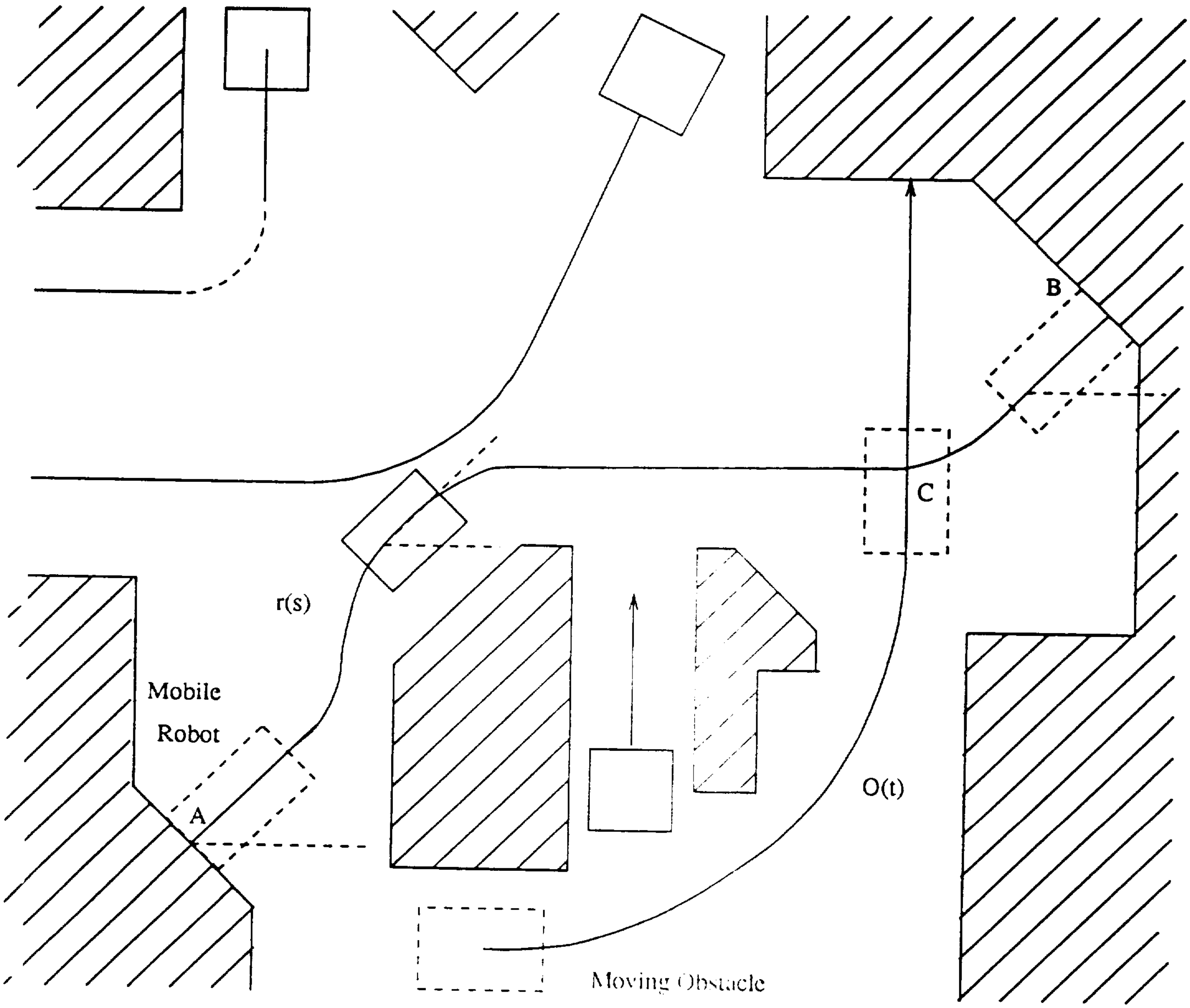

Fig. 1 Environment with Multiple Moving Objects

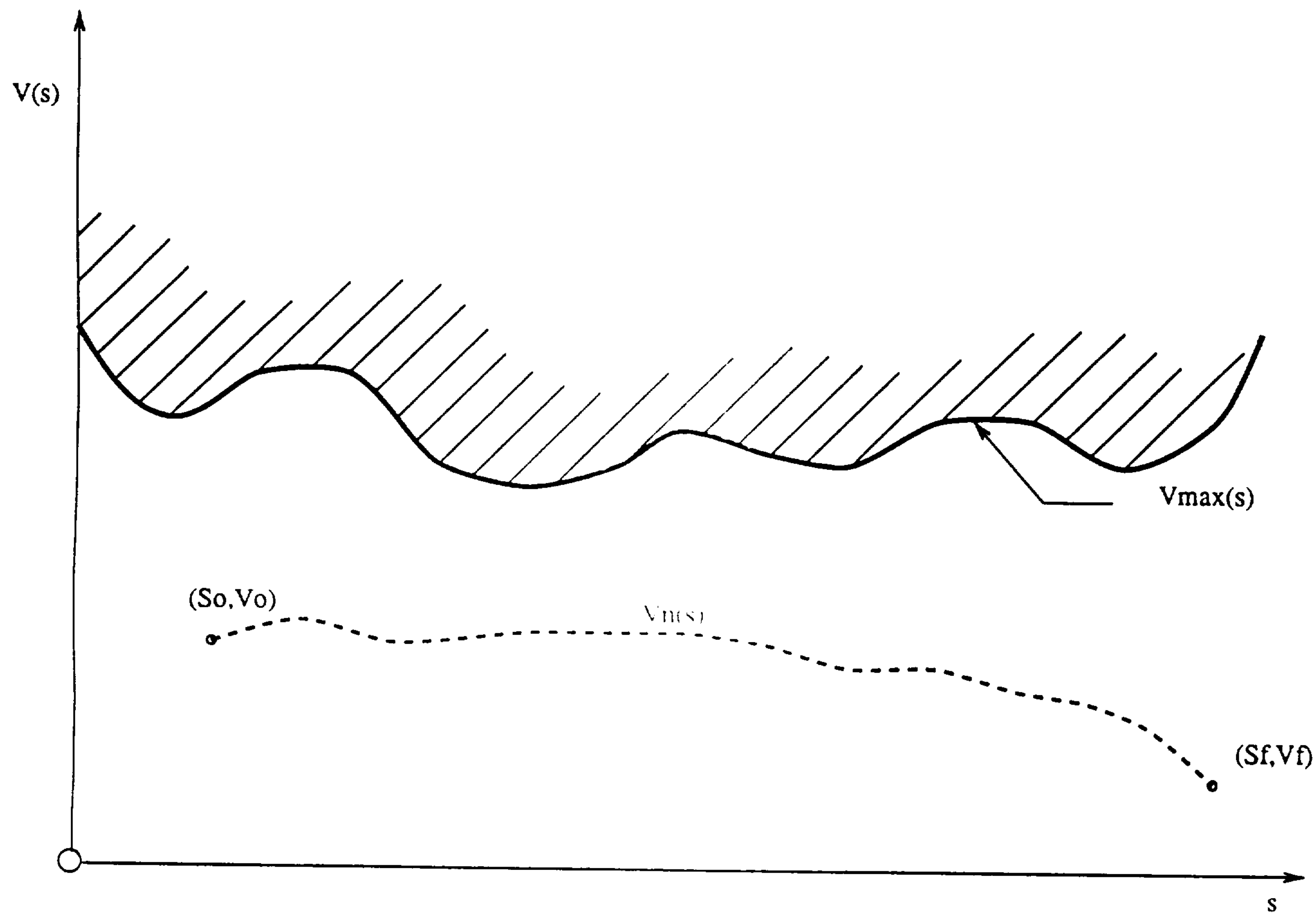

Fig. 2 Feasible Space and Nominal Velocity Arc

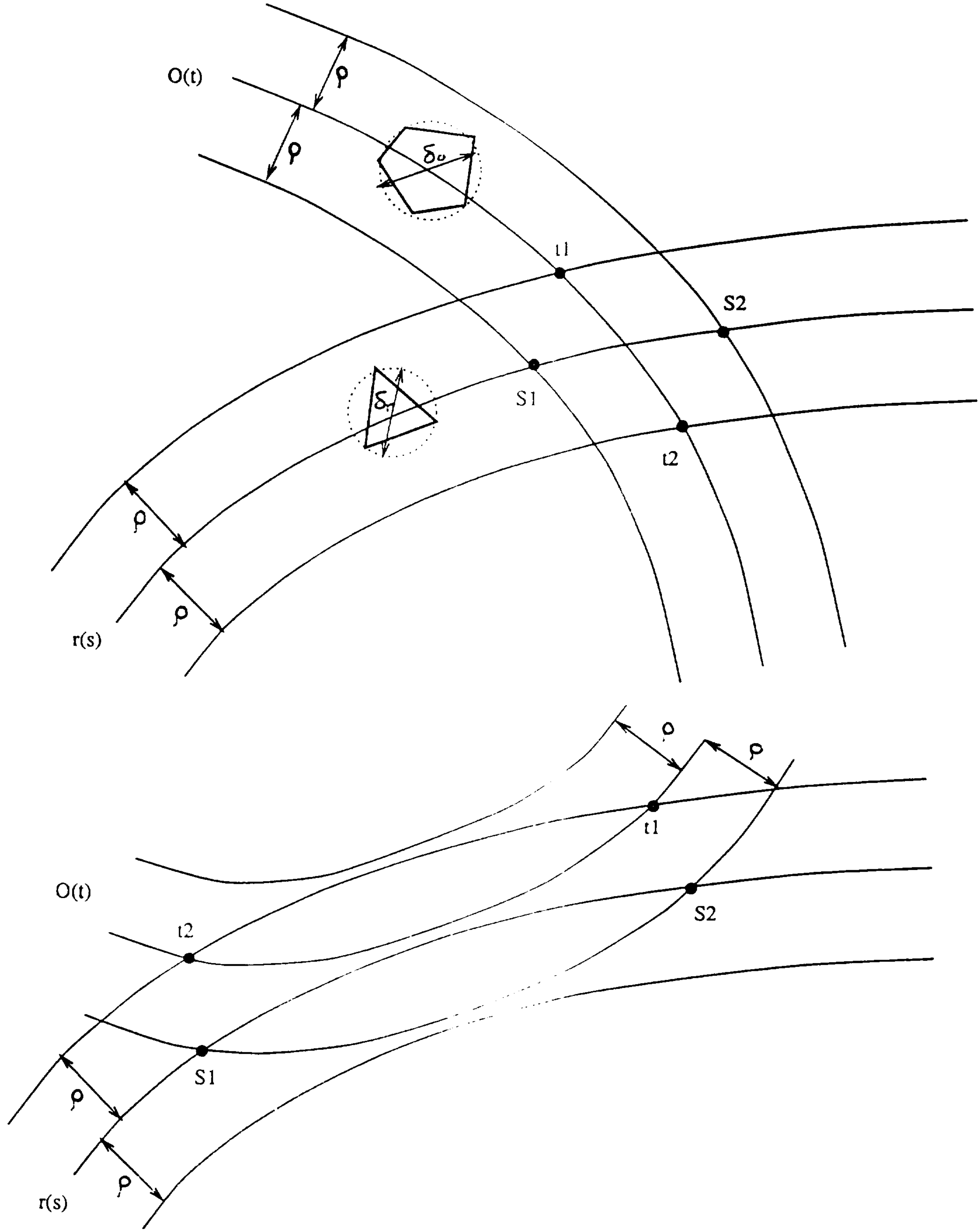

Fig. 3 Extruction of "Dangerous" Segment-time Parameters

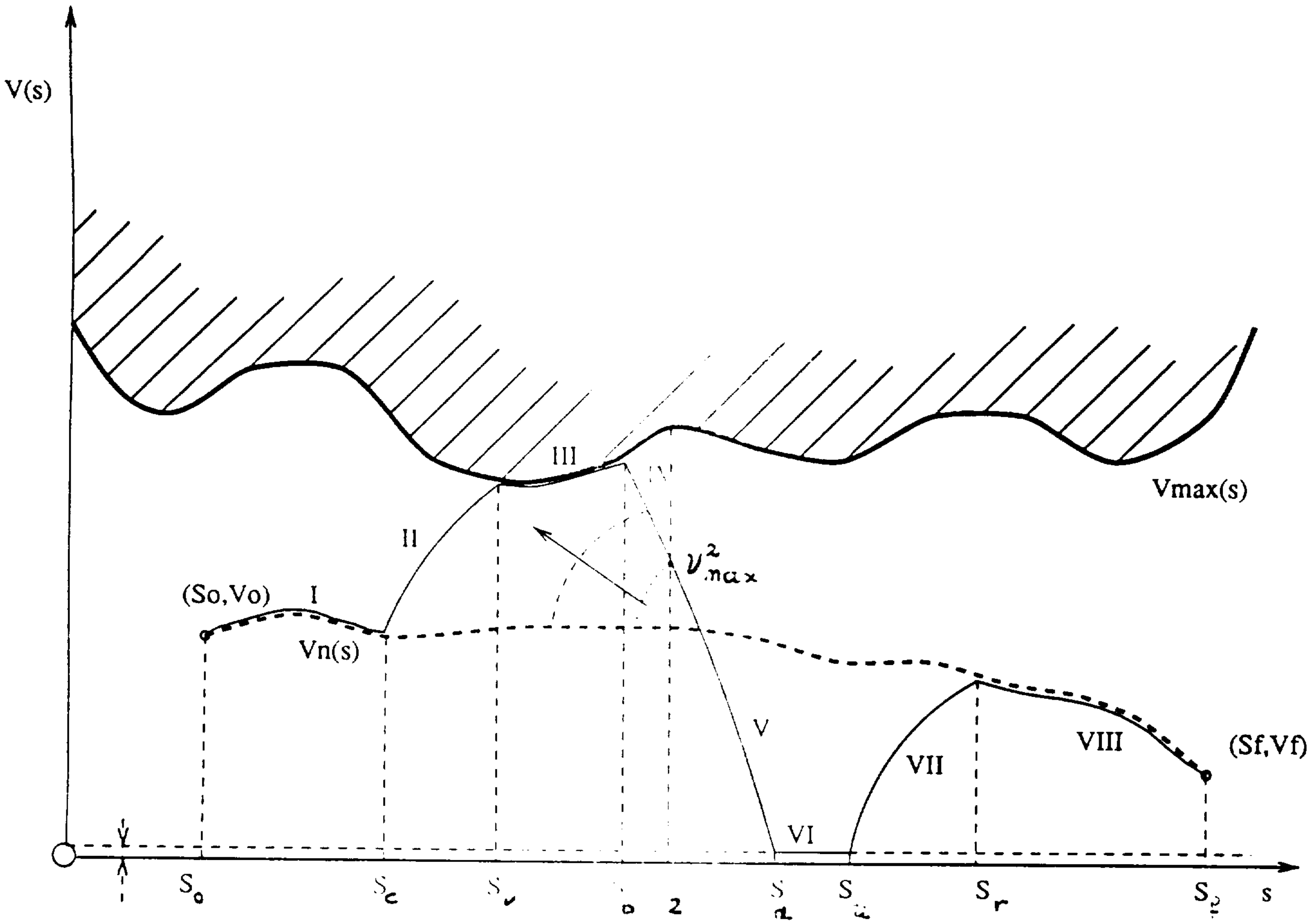

Fig 4. Convergence and Final Arc from AMIS

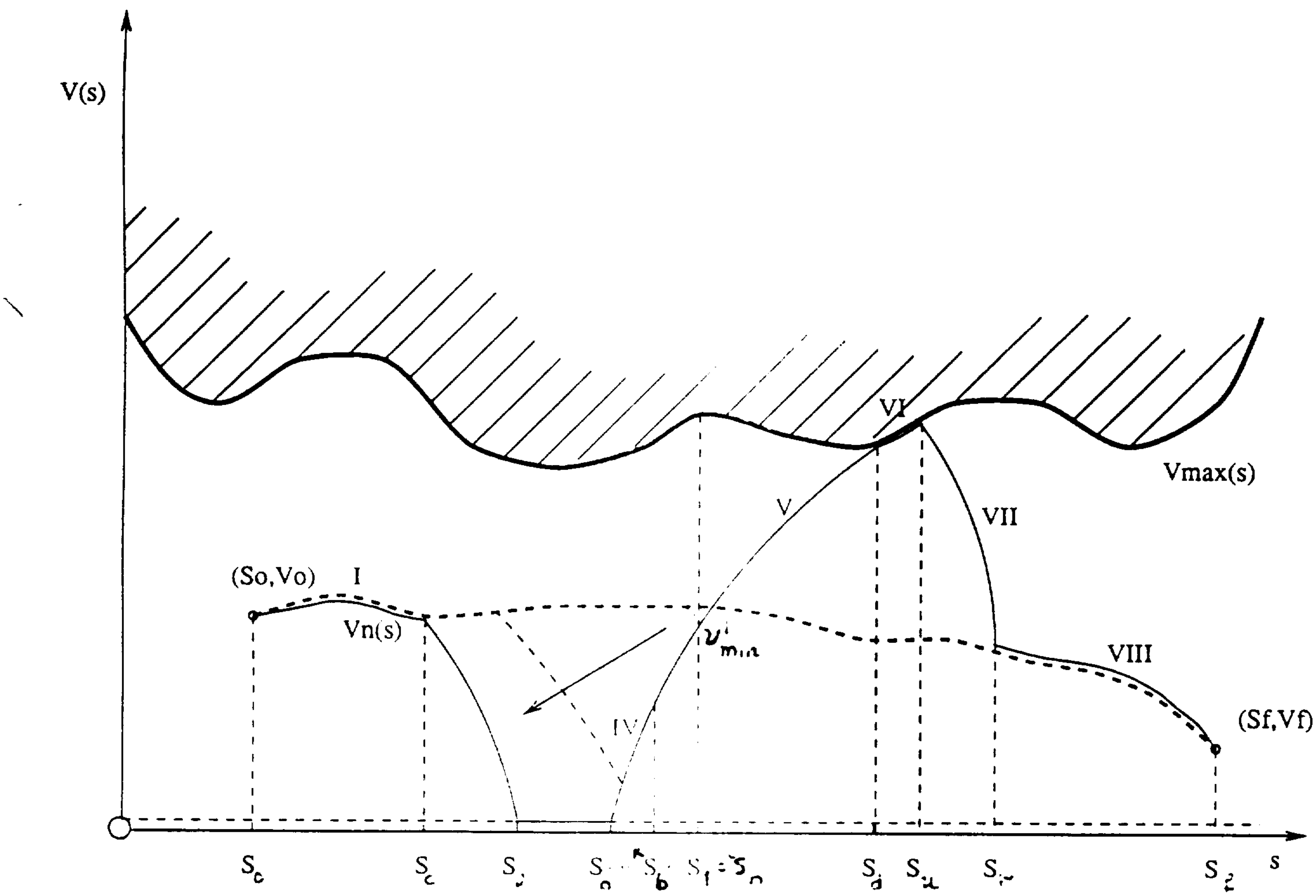

Fig. Convergence and Final Arc From DMIS

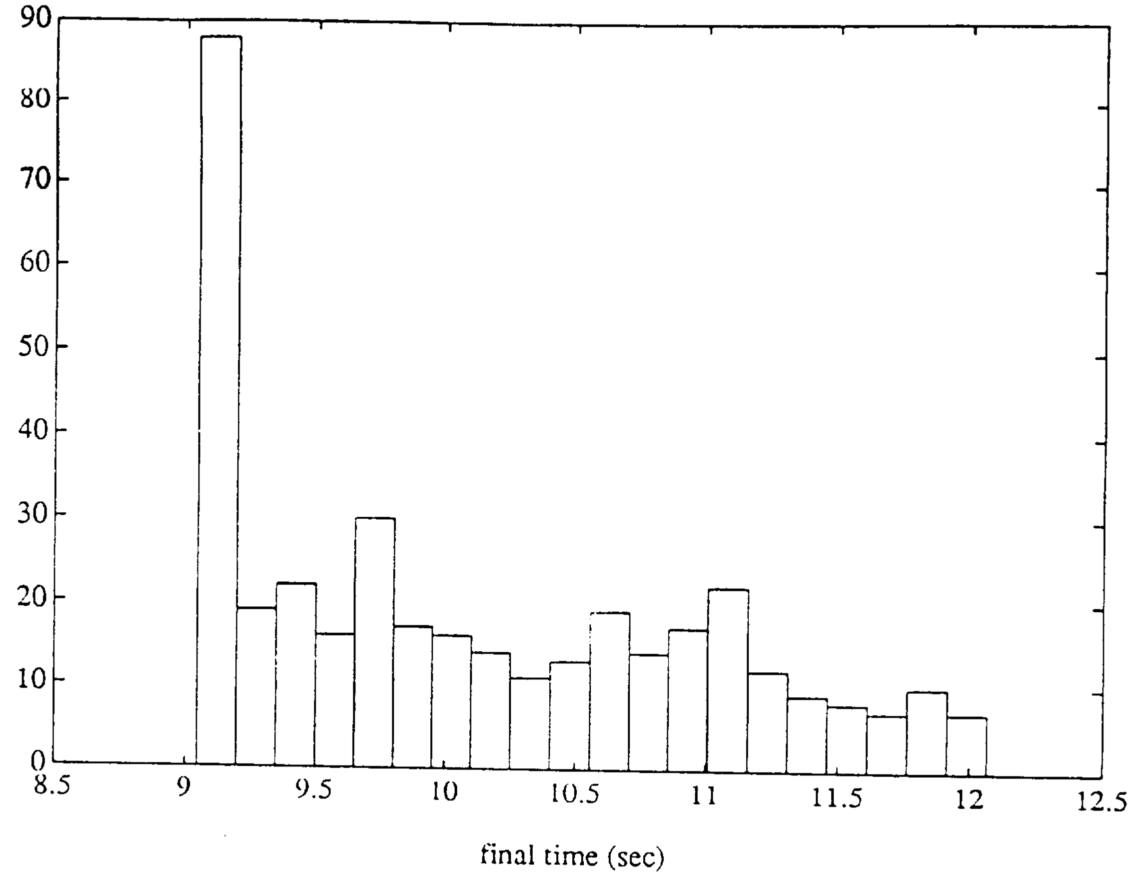

Fig. 6 Distribution of final time in collision avoidance cases.

Nominal time = 9.02 sec.

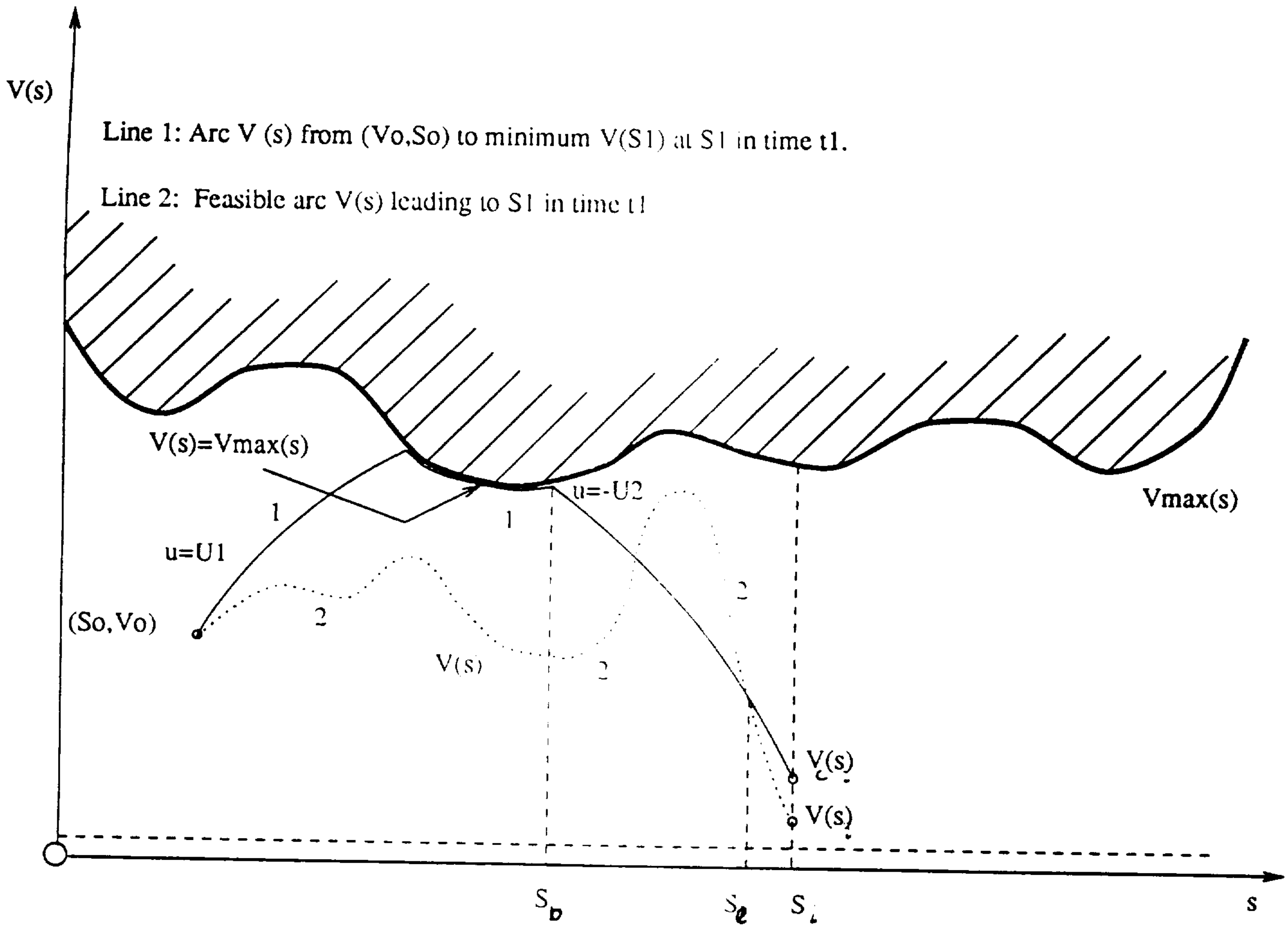

Fig. 7 Optimal and Suboptimal Velocity Arcs

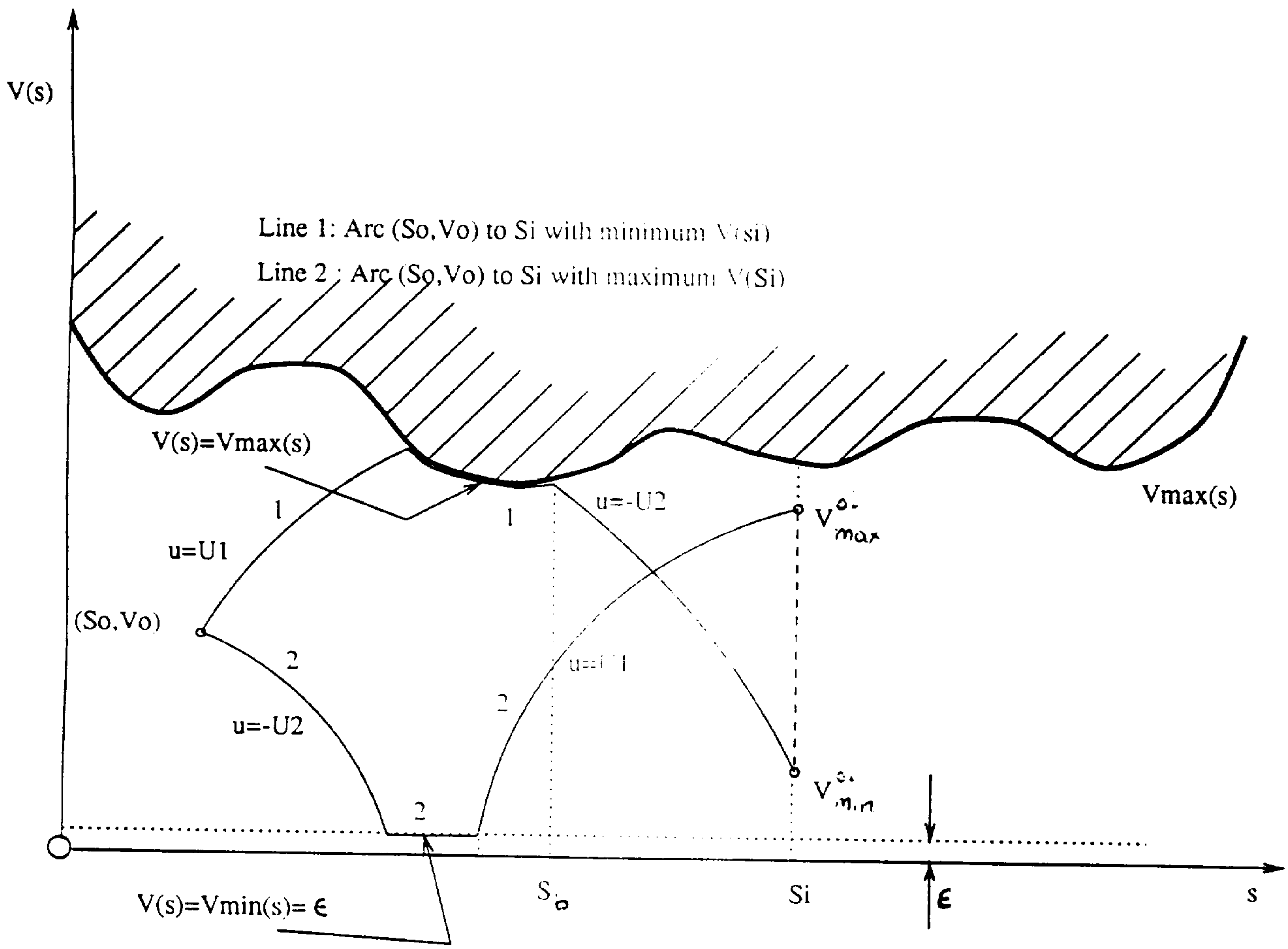

Fig. 8 Velocity Arcs Leading to $\dot{V}max$ and $\dot{V}min$

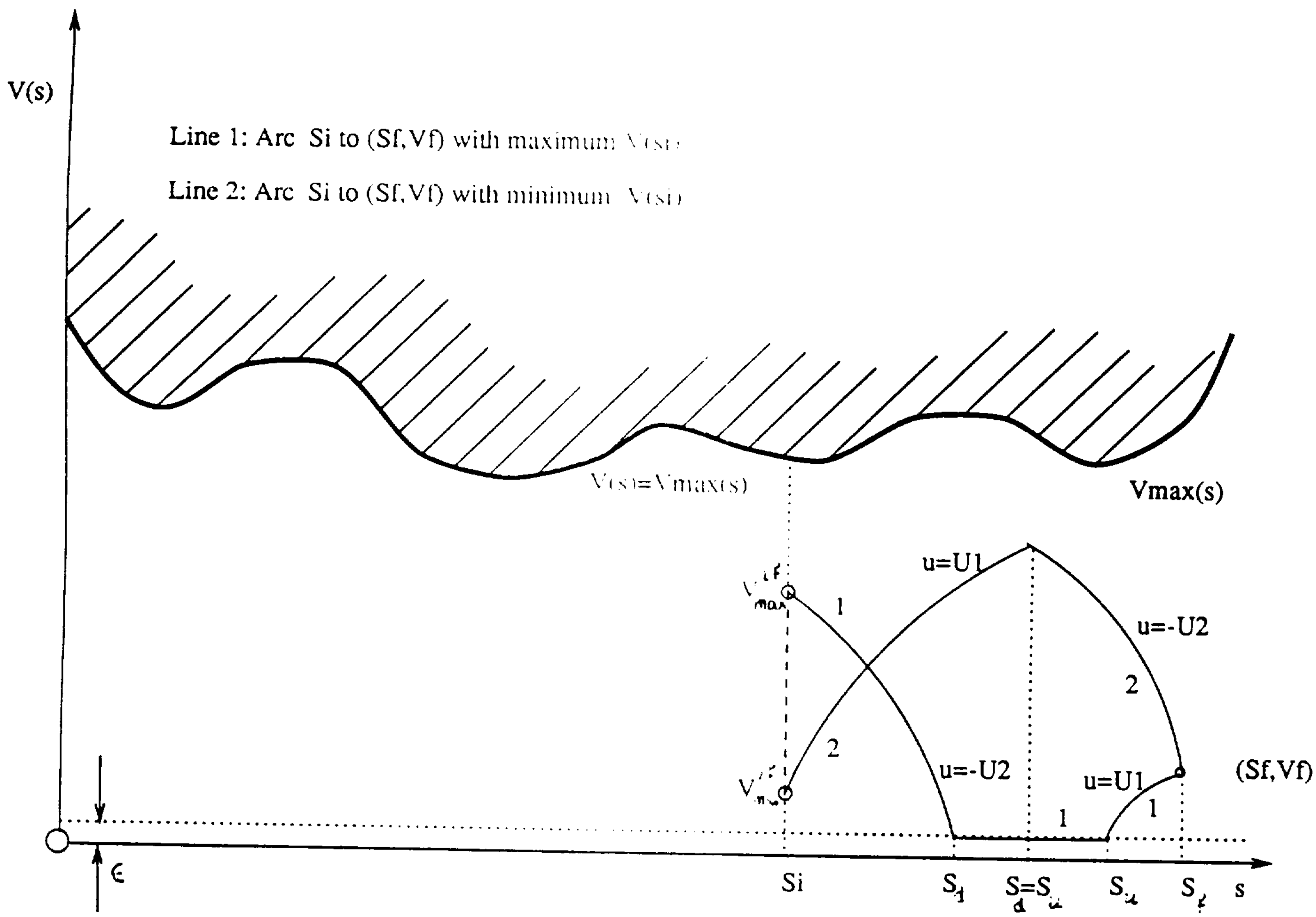

Fig. 9 Velocity Arcs Leading to Vmax and Vmin

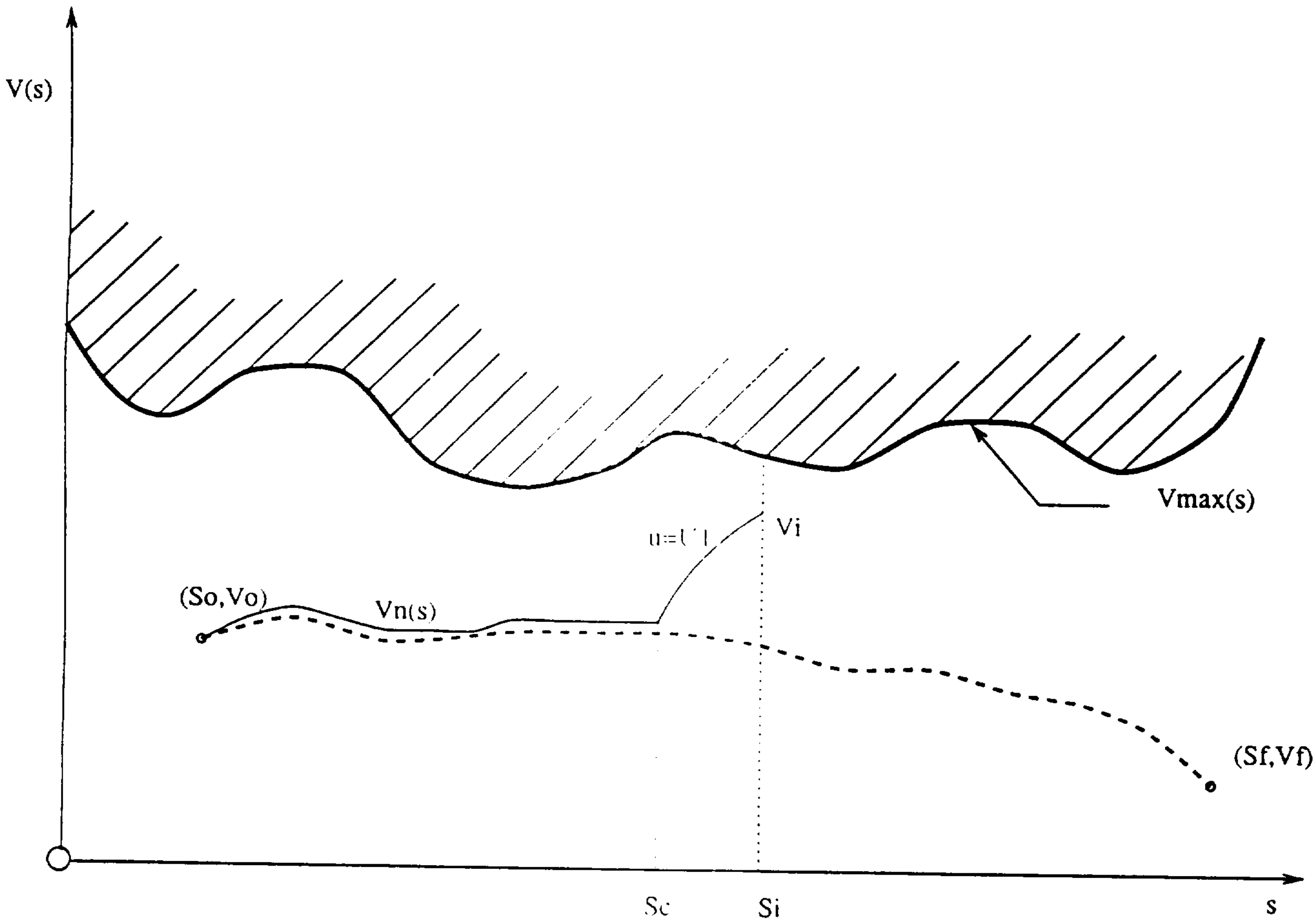

Fig. 10 Nominal Velocity Arc and en Intermediate Velocity Arc in AMIS